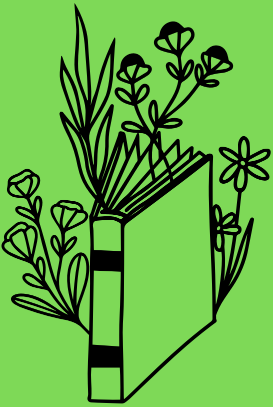
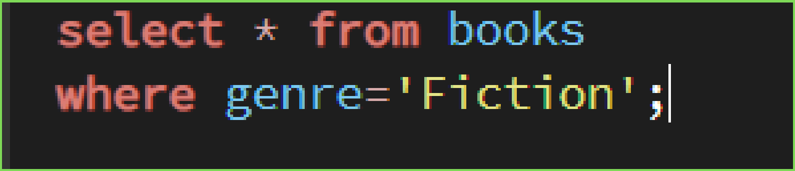


BOOK STORE

📚 Bookstore SQL Project

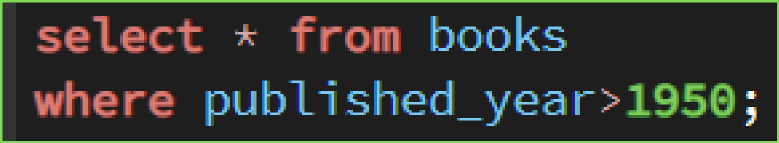
This project is a simple bookstore database built using PostgreSQL. It manages books, customers, and orders using three connected tables. The system tracks book inventory, customer details, and purchase history, making it easy to analyze sales, stock, and customer activity.

Retrieve all books in the "Fiction" genre



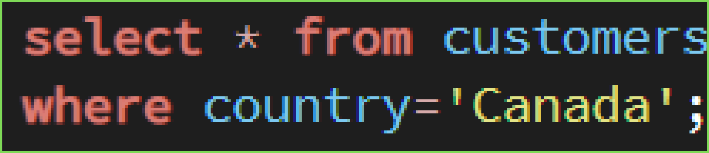
```
select * from books
where genre='Fiction';
```

Find books published after the year 1950



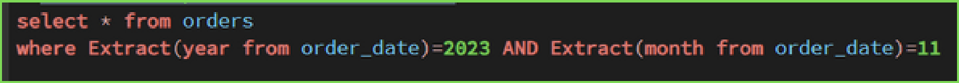
```sql
select * from books
where published_year>1950;
```

List all customers from the Canada



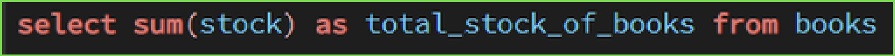
```sql
select * from customers
where country='Canada';
```

Show orders placed in November 2023



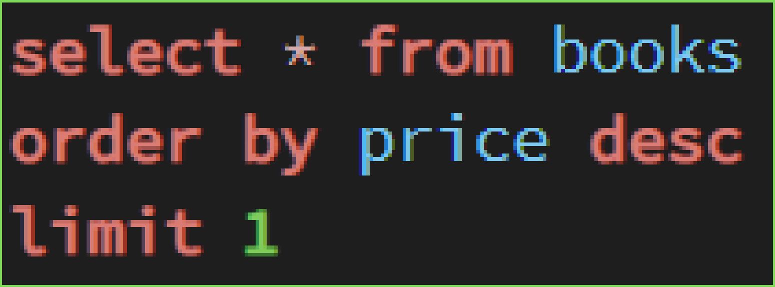
```
select * from orders
where Extract(year from order_date)=2023 AND Extract(month from order_date)=11
```

Retrieve the total stock of books available



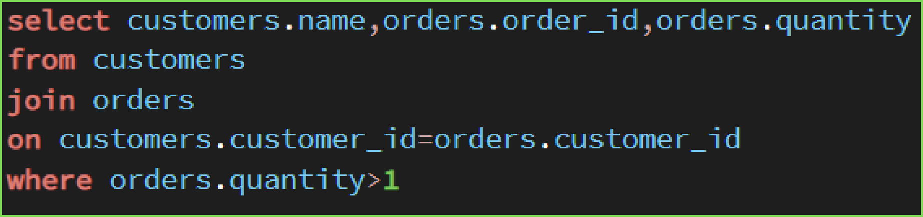
```sql
select sum(stock) as total_stock_of_books from books
```

Find the details of the most expensive book



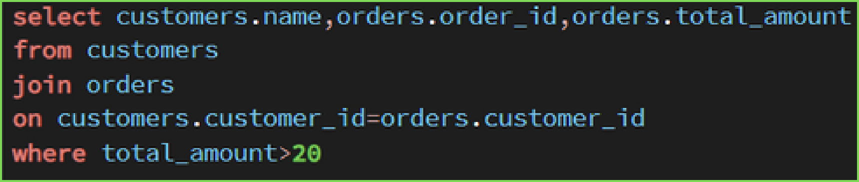
```
select * from books
order by price desc
limit 1
```

Show all customers who ordered more than 1 quantity of a book



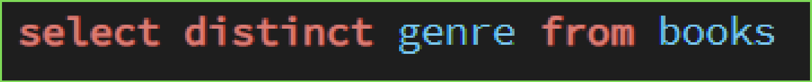
```
select customers.name,orders.order_id,orders.quantity
from customers
join orders
on customers.customer_id=orders.customer_id
where orders.quantity>1
```

Retrieve all orders where the total amount exceeds $20



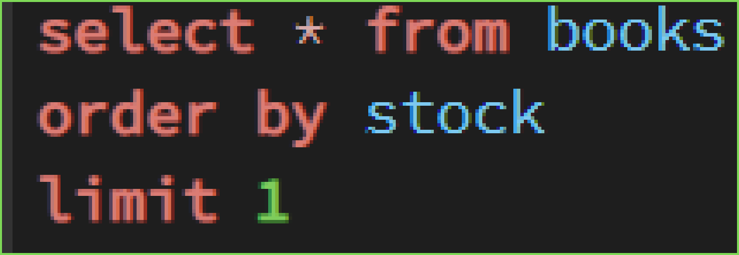
```
select customers.name,orders.order_id,orders.total_amount
from customers
join orders
on customers.customer_id=orders.customer_id
where total_amount>20
```

List all genres available in the Books table



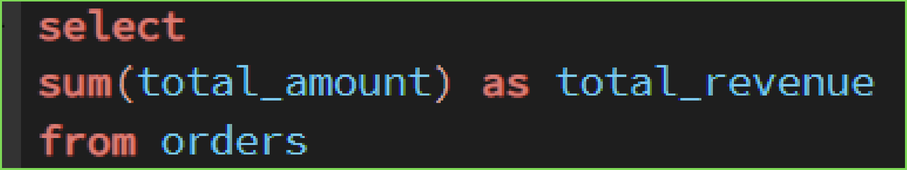
```
select distinct genre from books
```

Find the book with the lowest stock



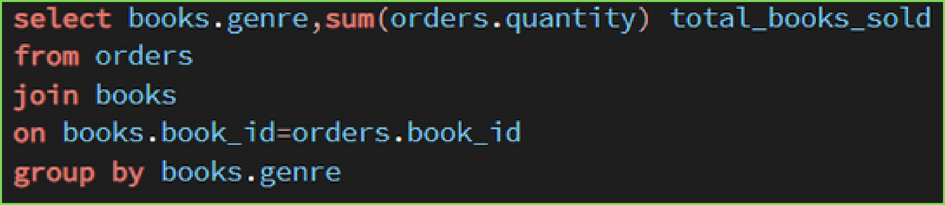
```
select * from books
order by stock
limit 1
```

Calculate the total revenue generated from all orders



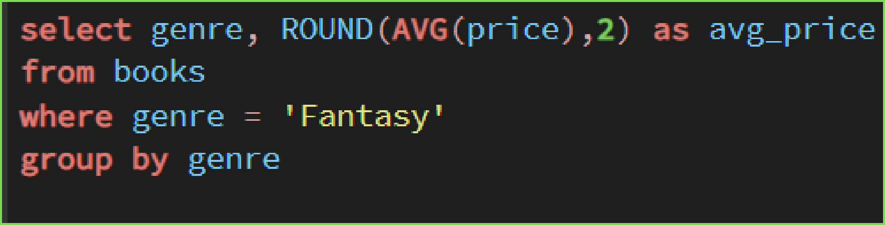
```sql
select
sum(total_amount) as total_revenue
from orders
```

Retrieve the total number of books sold for each genre



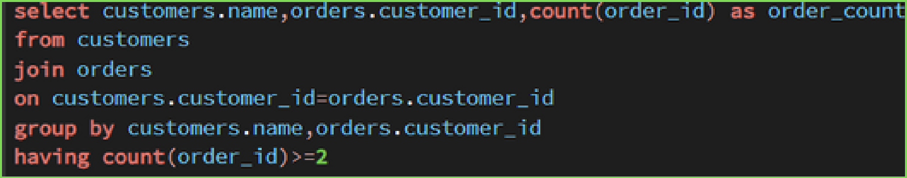
```
select books.genre,sum(orders.quantity) total_books_sold
from orders
join books
on books.book_id=orders.book_id
group by books.genre
```

Find the average price of books in the "Fantasy" genre



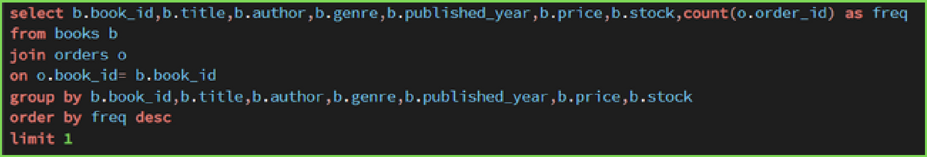
```sql
select genre, ROUND(AVG(price),2) as avg_price
from books
where genre = 'Fantasy'
group by genre
```

List customers who have placed at least 2 orders



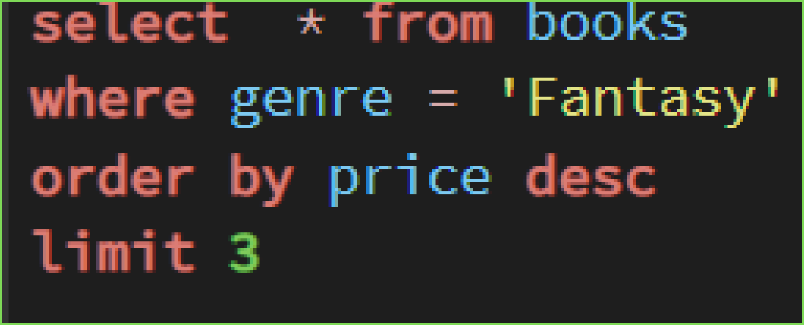
```
select customers.name,orders.customer_id,count(order_id) as order_count
from customers
join orders
on customers.customer_id=orders.customer_id
group by customers.name,orders.customer_id
having count(order_id)>=2
```

Find the most frequently ordered book



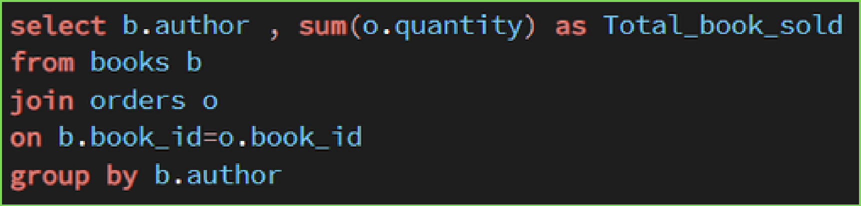
```sql
select b.book_id,b.title,b.author,b.genre,b.published_year,b.price,b.stock,count(o.order_id) as freq
from books b
join orders o
on o.book_id= b.book_id
group by b.book_id,b.title,b.author,b.genre,b.published_year,b.price,b.stock
order by freq desc
limit 1
```

Show the top 3 most expensive books of 'Fantasy' Genre



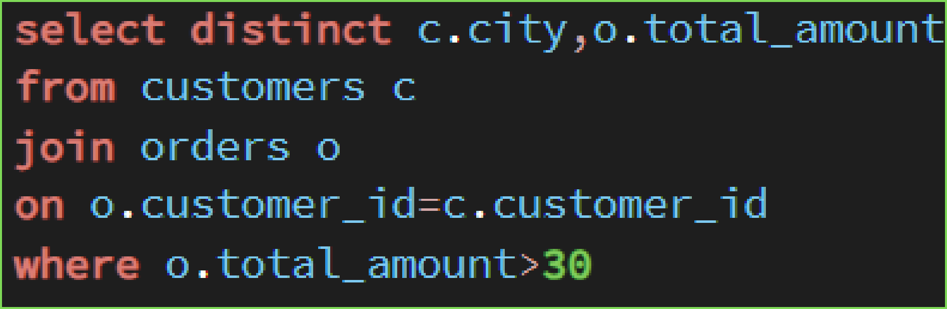
```
select  * from books
where genre = 'Fantasy'
order by price desc
limit 3
```

Retrieve the total quantity of books sold by each author



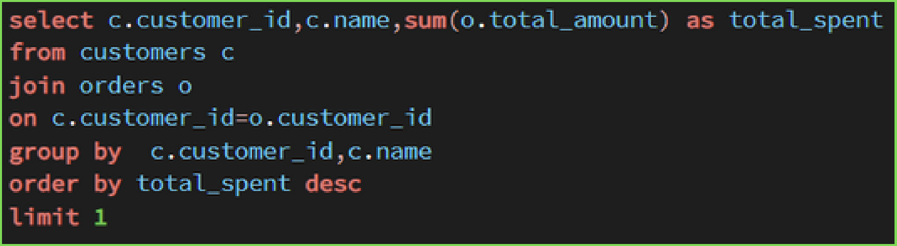
```
select b.author , sum(o.quantity) as Total_book_sold
from books b
join orders o
on b.book_id=o.book_id
group by b.author
```

List the cities where customers who spent over $30 are located



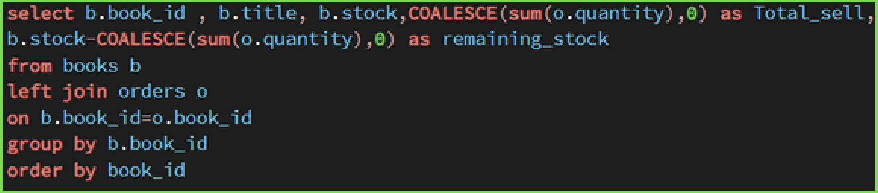
```
select distinct c.city,o.total_amount
from customers c
join orders o
on o.customer_id=c.customer_id
where o.total_amount>30
```

Find the customer who spent the most on orders



```
select c.customer_id,c.name,sum(o.total_amount) as total_spent
from customers c
join orders o
on c.customer_id=o.customer_id
group by  c.customer_id,c.name
order by total_spent desc
limit 1
```

Calculate the stock remaining after fulfilling all orders



```
select b.book_id , b.title, b.stock,COALESCE(sum(o.quantity),0) as Total_sell,
b.stock-COALESCE(sum(o.quantity),0) as remaining_stock
from books b
left join orders o
on b.book_id=o.book_id
group by b.book_id
order by book_id
```